COMPUTER SCIENCE

A. K. Watson Hall, 203.432.1246
http://cpsc.yale.edu
M.S., M.Phil., Ph.D.

Chair
Zhong Shao

Director of Graduate Studies
Vladimir Rokhlin (AKW 108, 203.432.1278, vladimir.rokhlin@yale.edu)


Associate Professors Abhishek Bhattacharjee, Theodore Kim, Sahand Negahban,* Ruzica Piskac, Jakub Szefer*

Assistant Professors Yang Cai, Wenjun Hu,* Julian Jara-Ettinger,* Amin Karbasi,* Smita Krishnaswamy,* Robert Soulé, David van Dijk,* Marynel Vázquez

Senior Lecturers James Glenn, Kyle Jensen,* Stephen Slade

Lecturers Andrew Bridy,† Benedict Brown, Cody Murphey, Scott Petersen, Brad Rosen, Andrew Sherman, Cecillia Xie

* A secondary appointment with primary affiliation in another department or school.
† A joint appointment with another department.

FIELDS OF STUDY
Algorithms and computational complexity, artificial intelligence, data networking, databases, graphics, machine learning, programming languages, robotics, scientific computing, security and privacy, and systems.

RESEARCH FACILITIES
The department operates a high-bandwidth, local-area computer network based mainly on distributed workstations and servers, with connections to worldwide networks. Workstations include Dell dual-processor PCs (running Linux or Windows/XP). Laboratory contains specialized equipment for graphics, vision, and robotics research. Various printers, including color printers, as well as image scanners, are also available. The primary educational facility consists of thirty-seven PC workstations supported by a large Intel PC server. This facility is used for courses and unsponsored research by Computer Science majors and first-year graduate students. Access to computing, through both the workstations and remote login facilities, is available to everyone in the department.

SPECIAL REQUIREMENTS FOR THE PH.D. DEGREE
There is no foreign language requirement. To be admitted to candidacy, a student must (1) pass ten courses (including CPSC 690 and CPSC 691) with at least two grades of Honors, the remainder at least High Pass, including three advanced courses in an area of specialization; (2) take six advanced courses in areas of general computer science; (3) successfully complete a research project in CPSC 690, CPSC 691, and submit a written report on it to the faculty; (4) pass a qualifying examination in an area of specialization; (5) be accepted as a thesis student by a regular department faculty member; (6) serve as a teaching assistant for two terms at a TF level 10; and (7) submit a written dissertation prospectus, with a tentative title for the dissertation. To satisfy the distribution requirement (requirement 2 above), the student must take one course in programming languages or systems, one programming-intensive course, two theory courses, and two in application areas. In order to gain teaching experience, all graduate students are required to serve as teaching assistants for two terms during their first three years of study. All requirements for admission to candidacy must be completed prior to the end of the third year. In addition to all other requirements, students must successfully complete CPSC 991, Ethical Conduct of Research, prior to the end of their first year of study. This requirement must be met prior to registering for a second year of study.

MASTER’S DEGREES
M.Phil. See Degree Requirements under Policies and Regulations.

M.S. (en route to the Ph.D.) To qualify for the M.S., the student must pass eight courses at the 500 level or above from an approved list. An average grade of at least High Pass is required, with at least one grade of Honors.

Terminal Master's Degree Program Students may also be admitted to a terminal master's degree program directly. The requirements are the same as for the M.S. en route to the Ph.D. This program is normally completed in one year, but a part-time program may be spread over as many as four years.
A brochure providing additional information about the department, faculty, courses, and facilities is available from the Graduate Coordinator, Department of Computer Science, Yale University, PO Box 208285, New Haven CT 06520-8285; e-mail, cs-admissions@cs.yale.edu.

COURSES

**CPSC 521b, Compilers and Interpreters**  Robert Soule  
Compiler organization and implementation: lexical analysis, formal syntax specification, parsing techniques, execution environment, storage management, code generation and optimization, procedure linkage, and address binding. The effect of language-design decisions on compiler construction.

**CPSC 522a, Operating Systems**  Zhong Shao  
The design and implementation of operating systems. Topics include synchronization, deadlocks, process management, storage management, file systems, security, protection, and networking.

**CPSC 523b, Principles of Operating Systems**  Avi Silberschatz  
A survey of the underlying principles of modern operating systems. Topics include process management, memory management, storage management, protection and security, distributed systems, and virtual machines. Emphasis on fundamental concepts rather than implementation.

**CPSC 524b, Parallel Programming Techniques**  Andrew Sherman  
Practical introduction to parallel programming, emphasizing techniques and algorithms suitable for scientific and engineering computations. Aspects of processor and machine architecture. Techniques such as multithreading, message passing, and data parallel computing using graphics processing units. Performance measurement, tuning, and debugging of parallel programs. Parallel file systems and I/O.

**CPSC 525b, Mobile and Embedded Systems**  
Mobile and embedded systems are computers that are portable, embedded in a larger system, or both. They are usually resource constrained; intimately interact with the physical environment, including human users; and often serve mission-critical or privacy-sensitive applications. This course provides a comprehensive introduction to the inner workings of modern mobile and embedded systems, from hardware architecture to operating systems to algorithms. While the lectures focus on theory, principle, and even historical lessons, significant learning of practical systems hacking skills, including learning itself, come from six programming assignments, involving Linux kernel development, FreeRTOS, and bare metal systems. Prerequisite: CPSC 323.

**CPSC 531a, Computer Music: Algorithmic and Heuristic Composition**  Scott Petersen  
Study of the theoretical and practical fundamentals of computer-generated music. Music and sound representations, acoustics and sound synthesis, scales and tuning systems, algorithmic and heuristic composition, and programming languages for computer music. Theoretical concepts are supplemented with pragmatic issues expressed in a high-level programming language.

**CPSC 532b, Computer Music: Sound Representation and Synthesis**  Scott Petersen  
Study of the theoretical and practical fundamentals of computer-generated music, with a focus on low-level sound representation, acoustics and sound synthesis, scales and tuning systems, and programming languages for computer music generation. Theoretical concepts are supplemented with pragmatic issues expressed in a high-level programming language. Prerequisite: ability to read music.

**CPSC 533b, Computer Networks**  Anurag Khandelwal  
An introduction to the design, implementation, analysis, and evaluation of computer networks and their protocols. Topics include layered network architectures, applications, transport, congestion, routing, data link protocols, local area networks, performance analysis, multimedia networking, network security, and network management. Emphasis on protocols used in the Internet.

**CPSC 534b, Topics in Networked Systems**  Y. Richard Yang  
Study of networked systems such as the Internet and mobile networks which provide the major infrastructure components of an information-based society. Topics include the design principles, implementation, and practical evaluation of such systems in new settings, including cloud computing, software-defined networking, 5G, Internet of things, and vehicular networking.

**CPSC 535a, Internet-Scale Applications**  Robert Soule  
An introduction to the design and implementation of Internet-scale applications and services. Topics include service-oriented software design; cloud computing paradigms; infrastructure scalability and reliability; adaptive, open clients; protocol specification; performance modeling; debugging and diagnosis; and deployment and licensing. After CPSC 523a.

**CPSC 537a, Introduction to Database Systems**  Avi Silberschatz  

**CPSC 539b, Software Engineering**  Staff  
Introduction to building a large software system in a team. Learning how to collect requirements and write a specification. Project planning and system design. Increasing software reliability: debugging, automatic test generation. Introduction to type systems, static analysis, and model checking.
CPSC 546a, Data and Information Visualization  Holly Rushmeier and Benedict Brown
Visualization is a powerful tool for understanding data and concepts. This course provides an introduction to the concepts needed to build new visualization systems, rather than to use existing visualization software. Major topics are abstracting visualization tasks, using visual channels, spatial arrangements of data, navigation in visualization systems, using multiple views, and filtering and aggregating data. Case studies to be considered include a wide range of visualization types and applications in humanities, engineering, science, and social science. Prerequisite: CPSC 223.

CPSC 552a / AMTH 553a / CB&B 555a / GENE 555a, Unsupervised Learning for Big Data  Smita Krishnaswamy
This course focuses on machine-learning methods well-suited to tackling problems associated with analyzing high-dimensional, high-throughput noisy data including: manifold learning, graph signal processing, nonlinear dimensionality reduction, clustering, and information theory. Though the class goes over some biomedical applications, such methods can be applied in any field. Prerequisites: knowledge of linear algebra and Python programming.

CPSC 554a, Software Analysis and Verification  Ruzica Piskac
Introduction to concepts, tools, and techniques used in the formal verification of software. State-of-the-art tools used for program verification; detailed insights into algorithms and paradigms on which those tools are based, including model checking, abstract interpretation, decision procedures, and SMT solvers.

CPSC 555a, Economics and Computation  Yang Cai
A mathematically rigorous investigation of the interplay of economic theory and computer science, with an emphasis on the relationship of incentive-compatibility and algorithmic efficiency. Particular attention to the formulation and solution of mechanism-design problems that are relevant to data networking and Internet-based commerce.

CPSC 556b / ENAS 951b, Wireless Technologies and the Internet of Things  Wenjun Hu
Fundamental theory of wireless communications and its application explored against the backdrop of everyday wireless technologies such as WiFi and cellular networks. Channel fading, MIMO communication, space-time coding, opportunistic communication, OFDM and CDMA, and the evolution and improvement of technologies over time. Emphasis on the interplay between concepts and their implementation in real systems. The labs and homework assignments require Linux and MATLAB skills and simple statistical and matrix analysis (using built-in MATLAB functions).

CPSC 557b, Sensitive Information in a Wired World  Michael Fischer
Issues of ownership, control, privacy, and accuracy of the huge amount of sensitive information about people and organizations that is collected, stored, and used by today’s ubiquitous information systems. Readings consist of research papers that explore both the power and the limitations of existing privacy-enhancing technologies such as encryption and "trusted platforms."

CPSC 559a, Building Interactive Machines  Marynel Vazquez
This advanced course brings together methods from machine learning, computer vision, robotics, and human-computer interaction to enable interactive machines to perceive and act in a variety of environments. Part of the course examines approaches for perception with different sensing devices and algorithms; the other part focuses on methods for decision-making and applied machine learning for control. The course is a combination of lectures, state-of-the-art reading, presentations and discussions, programming assignments, and a final team project. Prerequisites: CPSC 570 and understanding of probability, differential calculus, linear algebra, and planning (in Artificial Intelligence). Programming assignments require proficiency in Python and high-level familiarity with C++. Students who do not fit this profile may be allowed to enroll with the permission of the instructor.

CPSC 560a, Automata Theory and Formal Languages  Andrew Bridy
Introduction to the theory of automata and formal languages, one of the building blocks of theoretical computer science. Major topics covered are finite automata, pushdown automata, and Turing machines, and their associated languages.

CPSC 564b, Topics in Foundations of Machine Learning  Nisheeth Vishnoi
This course focuses on current and important topics in machine learning where a foundational understanding is lacking or under development. This includes modern algorithmic methods, novel learning and generative models, and the societal impact of machine learning. Representative topics include optimization and learning methods for nonconvex functions in Euclidean spaces or manifolds, algorithms beyond worst case, fairness, and robustness. The course is for students who would like to address the limitations of current machine learning systems deployed in the real world through a combination of foundational work such as coming up with the right definitions, modeling, methods, along with empirical evaluation. The grade will be based on class participation and project. Project grade will be determined by a midterm and endterm report/presentation. The course has four primary modules, each roughly three to four weeks: (1) Introduction. Methods: overview of continuous optimization methods; overview of sampling methods; advanced methods such as minimax optimization, optimization, and sampling on manifolds; (2) Models: traditional models in supervised and unsupervised learning; maximum entropy-based generative models; neural networks, convolutional neural networks, and generative adversarial networks; (3) Robustness in ML: adversarial examples and misclassification; notions of robustness; methods for robust training; (4) Fairness in ML: sociotechnical contexts and the underlying algorithmic/ML problems; definitions of fairness; methods/models/interventions for fair ML; implicit bias and downstream effects of interventions. Prerequisites: solid background in calculus, linear algebra, stochastic processes, and advanced algorithms along with a good background in programming. CPSC 365 or CPSC 366 is required, and S&DS 251 is recommended.
CPSC 565a, Theory of Distributed Systems  James Aspnes
Models of asynchronous distributed computing systems. Fundamental concepts of concurrency and synchronization, communication, reliability, topological and geometric constraints, time and space complexity, and distributed algorithms.

CPSC 567a, Cryptography and Computer Security  Michael Fischer
A survey of such private and public key cryptographic techniques as DES, RSA, and zero-knowledge proofs, and their application to problems of maintaining privacy and security in computer networks. Focus on technology, with consideration of such societal issues as balancing individual privacy concerns against the needs of law enforcement, vulnerability of societal institutions to electronic attack, export regulations and international competitiveness, and development of secure information systems.

CPSC 569b, Randomized Algorithms  James Aspnes
Beginning with an introduction to tools from probability theory including some inequalities like Chernoff bounds, the course covers randomized algorithms from several areas: graph algorithms, algorithms in algebra, approximate counting, probabilistically checkable proofs, and matrix algorithms.

CPSC 570b, Artificial Intelligence  Brian Scassellati
Introduction to artificial intelligence research, focusing on reasoning and perception. Topics include knowledge representation, predicate calculus, temporal reasoning, vision, robotics, planning, and learning.

CPSC 574a, Computational Intelligence for Games  James Glenn

CPSC 575a / ENAS 575a, Computational Vision and Biological Perception  Steven Zucker
An overview of computational vision with a biological emphasis. Suitable as an introduction to biological perception for computer science and engineering students, as well as an introduction to computational vision for mathematics, psychology, and physiology students.

CPSC 576b / AMTH 667b / ENAS 576b, Advanced Computational Vision  Steven Zucker
Advanced view of vision from a mathematical, computational, and neurophysiological perspective. Emphasis on differential geometry, machine learning, visual psychophysics, and advanced neurophysiology. Topics include perceptual organization, shading, color, and texture.

CPSC 577b, Natural Language Processing  Dragomir Radev
Linguistic, mathematical, and computational fundamentals of natural language processing (NLP). Topics include part of speech tagging, Hidden Markov models, syntax and parsing, lexical semantics, compositional semantics, machine translation, text classification, discourse, and dialogue processing. Additional topics such as sentiment analysis, text generation, and deep learning for NLP.

CPSC 578b, Computer Graphics  Theodore Kim
Introduction to the basic concepts of two- and three-dimensional computer graphics. Topics include affine and projective transformations, clipping and windowing, visual perception, scene modeling and animation, algorithms for visible surface determination, reflection models, illumination algorithms, and color theory.

CPSC 579a, Advanced Topics in Computer Graphics  Julie Dorsey
An in-depth study of advanced algorithms and systems for rendering, modeling, and animation in computer graphics. Topics vary and may include reflectance modeling, global illumination, subdivision surfaces, NURBS, physically based fluids systems, and character animation.

CPSC 584b, Introduction to Human-Computer Interaction  Marynel Vazquez
This course introduces students to the interdisciplinary field of human-computer interaction (HCI), with particular focus on human-robot interaction (HRI). The first part of the course covers principles and techniques in the design, development, and evaluation of interactive systems. It provides students with an introduction to UX design and user-centered research. The second part focuses on the emergent filed of HRI and several other nontraditional interfaces, e.g., AR/VR, tangibles, crowdsourcing. The course is organized as a series of lectures, presentations, a midterm exam, and a term-long group project on designing a new interactive system. Prerequisites: CPSC 201 and CPSC 202 or equivalents. Students who do not fit this profile may be allowed to enroll with permission of the instructor.

CPSC 586b, Topics in Computer Science and Law  Joan Feigenbaum
This course focuses on socio-technical problems in computing, i.e., problems that cannot be solved through technological progress alone but rather require legal, political, or cultural progress as well. Examples include but are not limited to computer security, intellectual property protection, cyber crime, cyber war, surveillance, and online privacy. The course is addressed to graduate students in Computer Science who are interested in socio-technical issues but whose undergraduate work may not have addressed them; it is designed to bring these students rapidly to the point at which they can do research on socio-technical problems. Students do term projects (either papers or software artifacts) and present them at the end of the term. In order to ensure that there is enough time for both midterm feedback on project proposals and in-class presentation of the finished projects, enrollment is limited to fifteen. If fewer than fifteen Computer Science graduate students enroll, Yale College undergraduates will be allowed to enroll with permission of the instructor. Prerequisites: the basics of cryptography and computer security (as covered in CPSC 467), networks (as covered in CPSC 433), and databases (as covered in CPSC 437), or permission of the instructor.

CPSC 622b, Advanced Operating Systems  Abhishek Bhattacharjee
This course focuses on advanced topics in operating systems. In particular, topics such as microkernels and exokernels, virtual memory, memory protection, single-address space operating systems, system calls and the hardware interface, OS support for emerging
Accelerators (e.g., GPUs), and storage (including emerging byte-addressable non-volatile memories) are covered. Students are asked to read and present research papers from top-tier operating systems and computer architecture venues. Furthermore, students undertake a term-long research project on building a substantial layer in a production operating system. This course is aimed at Ph.D. and M.S. students and, with permission of the instructor, undergraduates, with a background in operating systems, computer architecture and organization, and systems programming. In particular, students must have successfully passed a systems programming and computer organization course (e.g., CPSC 323) and an operating systems course (e.g., CPSC 522 and/or CPSC 523). Exposure to other systems classes (e.g., CPSC 524, CPSC 533, CPSC 534, CPSC 536, CPSC 537) is useful but not required. Because the course is research-intensive, prerequisites are strictly enforced.

**CPSC 626a, Human Factors in Computer Systems: Design, Evaluation, and Presentation**

Humans are stupid; computers are limited. Yet a collaboration of humans and computers has led to ever more powerful and complex computer systems. This course examines the mental limitations of human users and developers of computers and how they shape the design, implementation, evaluation, and presentation of computer systems. The lectures, reading assignments, and classroom discussions travel through psychology and philosophy and revisit important results from theoretical computer science, with a goal of elucidating the rationales behind the best practices in computer systems research and development. Prerequisite: CPSC 522, CPSC 523, or equivalent.

**CPSC 637a, Big Data** Anurag Khandelwal

Today's Internet-scale applications and cloud services generate massive amounts of data. At the same time, the availability of inexpensive storage has made it possible for these services and applications to collect and store every piece of data they generate, in the hopes of improving their services by analyzing the collected data. This introduces interesting new opportunities and challenges designing systems for collecting, analyzing, and serving the so-called big data. This course looks at technology trends that have paved the way for big data applications, surveys state-of-the-art systems for storage and processing of big data, and considers future research directions driven by open research problems. It includes a mix of lectures, seminar-style discussions, student presentations, and a course-long research project. Readings are selected from top-tier conferences such as SOSP, OSDI, NSDI, SIGCOMM, SIGMOD, and EuroSys, spanning topics such as cluster architecture, big data analytics stacks, scheduling and resource management, machine learning, batch and stream analytics, graph processing, serverless platforms, and disaggregated architectures.

**CPSC 640b / AMTH 640b, Topics in Numerical Computation** Vladimir Rokhlin

This course discusses several areas of numerical computing that often cause difficulties to non-numericists, from the ever-present issue of condition numbers and ill-posedness to the algorithms of numerical linear algebra to the reliability of numerical software. The course also provides a brief introduction to “fast” algorithms and their interactions with modern hardware environments. The course is addressed to Computer Science graduate students who do not necessarily specialize in numerical computation; it assumes the understanding of calculus and linear algebra and familiarity with (or willingness to learn) either C or FORTRAN. Its purpose is to prepare students for using elementary numerical techniques when and if the need arises.

**CPSC 662a, Combinatorial Discrepancy and Its Applications** Staff

Combinatorial discrepancy asks the following question: Given a set and a family of its subsets, how do we two-color the set so that the coloring is nearly balanced in every subset? One naive idea is random coloring. Our goal is to improve upon random coloring. This course is an introduction to combinatorial discrepancy and covers many of its applications in computer science. The course has three parts: basic concepts and classical problems, efficient algorithms developed in the last decade, and the applications in randomized experimental designs, design of approximation algorithms, etc. Students learn elegant and powerful tools and techniques from linear algebra, combinatorics, and probability theory. Prerequisites: basic knowledge of linear algebra (at the level of MATH 222/223), probability and stochastic processes (mainly random walks, martingales, and concentration inequalities taught in S&DS 221/551), and algorithm design (taught in, for example, CPSC 366). Undergraduates need the instructor’s permission to enroll in the course.

**CPSC 674b, Advanced Computational Intelligence for Games** James Glenn

A seminar on current topics in computational intelligence for games, including developing agents for playing games, procedural content generation, and player modeling. Students read, present, and discuss recent papers and competitions, and complete a term-long project that applies some of the techniques discussed during the term to a game of their choice. Prerequisite: CPSC 574.

**CPSC 677a, Advanced Natural Language Processing** Dragomir Radev

Advanced topics in natural language processing (NLP), including related topics such as deep learning and information retrieval. Included are: (1) fundamental material not covered in the introductory NLP class such as text summarization, question answering, document indexing and retrieval, query expansion, graph-based techniques for NLP and IR, as well as (2) state-of-the-art material published in the past few years such as transfer learning, generative adversarial networks, reinforcement learning for NLP, sentence representations, capsule networks, multitask learning, and zero-shot learning. Prerequisite: CPSC 570, CPSC 577, or equivalent, or permission of the instructor.

**CPSC 678b, Creative Artificial Intelligence for Visual Computing** Julie Dorsey

How can artificial intelligence help us create visual content? In this readings-and-projects-based course, we explore how to use tools such as probabilistic models, probabilistic programs, and neural networks to generate content, explore design spaces, and support creativity for 2D and 3D graphics and vision applications. Each week, we read recent papers from the visual computing and AI literatures and discuss their contributions, connections, and limitations. Students also complete a collaborative, open-ended final project. Throughout, the course emphasizes key academic skills such as critical paper-reading and how to give clear and compelling presentations. Topics include arranging objects, controlling procedural models, generating 3D geometry, writing and drawing, assigning materials and colors,
generative adversarial models, learning representations of shape, learning from RGB-D panoramas, deep reinforcement learning, and exploring manifolds. Prerequisites: equivalent of CPSC 323 and some background in computer graphics, artificial intelligence, machine learning, or probabilistic modeling are helpful. Students are not expected to be familiar with all of these areas; this is a multidisciplinary area, and we welcome students of diverse backgrounds to share their expertise and interests. Students who lack this background may be allowed to enroll with permission of the instructor.

**CPSC 690a or b, Independent Project I**  Staff
By arrangement with faculty.

**CPSC 691a or b, Independent Project II**  Staff
By arrangement with faculty.

**CPSC 692a or b, Independent Project**  Staff
Individual research for students in the M.S. program. Requires a faculty supervisor and the permission of the director of graduate studies.

**CPSC 752b / CB&B 752b / MB&B 753b and MB&B 754b / MB&B 753b and MB&B 754b / MB&B 754b / MCDB 752b, Biomedical Data Science: Mining and Modeling**  Mark Gerstein and Matthew Simon
Biomedical data science encompasses the analysis of gene sequences, macromolecular structures, and functional genomics data on a large scale. It represents a major practical application for modern techniques in data mining and simulation. Specific topics to be covered include sequence alignment, large-scale processing, next-generation sequencing data, comparative genomics, phylogenetics, biological database design, geometric analysis of protein structure, molecular-dynamics simulation, biological networks, normalization of microarray data, mining of functional genomics data sets, and machine-learning approaches to data integration. Prerequisites: biochemistry and calculus, or permission of the instructor.

**CPSC 990a, Ethical Conduct of Research for Master's Students**  Holly Rushmeier
This course meets on four consecutive Fridays.

**CPSC 991a / MATH 991a, Ethical Conduct of Research**  Holly Rushmeier
- 0 Course cr